

**REMARKS**

Claims 1-51 are pending in the application. Claims 1, 2, 5, 12, 17-23, 25, 34-40 and 47 have been amended. No claims have been canceled. In view of the above amendments and these remarks, the application is believed to be in condition for allowance.

**Claim Objections**

Claims 1, 2, 5, 17-23, and 35-39 have been objected to due to informalities. Each of these claims has been amended to correct the informalities as suggested by the examiner. Specifically, the terminology "computer implemented method" is clearly broader in definition than "computer process" as originally submitted in claims 1, 2, 5, 17-23, and 35-39. The terminology has been changed as suggested by the examiner and is now broader than originally submitted. Thus the substitution of this terminology as the examiner suggested does not constitute a narrowing amendment of each of these claims which would invoke the prosecution history estoppel doctrine under the recent Supreme Court and Federal Circuit Festo decisions.

**Claim Rejections - 35 U.S.C. § 112**

Claim 12 stands rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Claim 12 has been amended, replacing the article "the" with --an--, to provide proper antecedent basis, thus this claim is now believed to be definite. This rejection should now be withdrawn.

**Claim Rejections - 35 U.S.C. § 102**

Claims 1-6, 15-26, 31, 32, and 34-51 stand rejected under 35 U.S.C. § 102(b) as being anticipated by Jagannathan et al. (USPN 5,692,193). As is set forth above, independent claims 1, 25, 34, 35 , 40, and 47 have been amended to recite that analysis and allocation of data to heaps specifically occurs during code compilation, prior to target program run time. Support for these claim amendments is specifically found in Applicant's specification, page 16, line 9 through page 17, line 4, and throughout the remainder of the specification.

Jagannathan et al discloses a system that utilizes a Sting compiler. The association of data with local or global heaps is determined at run time. Specifically, at column 10, lines 21-35, this patent states:

The Sting compiler is a modified version of Orbit. Orbit is described in an article by D. Kranz. et al, entitled "ORbit: An Optimizing Compiler for Scheme" in ACM SIGPLAN Notices, 21, (7): 219-233, July 1986. The target machine seen by the compiler includes a dedicated register to hold a reference to **the currently running thread object**. Moreover, time critical operations such as those that save and restore registers on a context switch, or allocate thread storage (i.e., stack and heaps) are provided as primitive operations. Sequential Scheme programs will compile and execute without modification. Sting implementations of futures, distributed data structures, and speculative concurrency operations also exist; Scheme programs can be freely augmented with the concurrency operations supported by any of these paradigms. (Emphasis added.)

Thus this patent teaches identification of threads and associated heaps, both local and global, as shown in Fig. 5, at run time, not during program compilation in advance of execution, or running, of a program. There is no disclosure or suggestion in this patent of performing thread specific and shared data analysis and allocations prior to running the target program as applicant now claims. Accordingly the rejection under 35 USC 102(b) should now be withdrawn.

### **Claim Rejections - 35 U.S.C. § 103**

Claims 7-14, 27-30, and 33 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Jaggannathan et al. in view of Benayon et al. (USPN 5,809,554). Benayon et al discloses a runtime library for implementing multiple heaps under programmer control. This differs substantially from applicant's claims wherein the compiler automatically determines which objects are local to a thread and which are shared and allocates those objects to a thread specific heap during target code compilation. Benayon et al does not teach identification of thread specific objects and allocation of such objects to thread specific heaps during compilation. Accordingly, Benayon et al cannot be combined with Jagannathan et al. to render claims 7-14, 27-30 and 33 obvious. In view of the amendments set forth above, this rejection should now be withdrawn.

### **Conclusion**

This amendment is believed to be responsive to all points raised in the Office Action.

Claims 1-51 remain in the application and are believed to be allowable for the reasons set forth

Application No. 09/761,404

above. As all claims now in the application are in condition for allowance, Applicant requests the application be allowed and pass to issuance as soon as possible.

It is believed that no further fees are due with this Response. However, the Commissioner is hereby authorized to charge any deficiencies or credit any overpayment with respect to this patent application to deposit account number 13-2725.

Date

6/30/04



Respectfully submitted,

A handwritten signature in black ink, appearing to read "John R. Wahl".  
John R. Wahl, Reg. No. 33,044  
Merchant & Gould P.C.  
P.O. Box 2903  
Minneapolis, MN 55402-0903  
(303) 357-1644  
(303) 357-1671 (fax)